Title

"Mirror Maze: Light Link"

A minimalist puzzle game where players redirect light beams to activate mirrors and unlock doors.

Core Concept

- **Objective**: Guide a beam of light from a fixed emitter to a receiver by rotating mirrors placed around the level.
- **Twist**: Mirrors only activate (become solid) when illuminated, creating a chain reaction. Players must plan the path so that all required mirrors are lit simultaneously to unlock the exit.
- **Looping Mechanic**: Each level resets if the player fails to light all mirrors within a set number of moves, encouraging strategic experimentation.

Gameplay Mechanics

1. Light Emitter and Receiver

- **Emitter**: A static actor that shoots a light beam in one direction (e.g., right).
- **Receiver**: A static actor that unlocks the exit door when the light beam hits it.
- **Light Beam**: A visual effect (animated sprite) that travels in a straight line until it hits a mirror or wall.

2. Rotatable Mirrors

- **Mirrors**: Actors with 4 rotation states (up, down, left, right). Players can interact with them to change their angle.
- **Activation**: Mirrors only reflect light when hit by the beam. Unlit mirrors are semi-transparent; lit mirrors become solid and reflective.
- **Chain Reaction**: Lighting one mirror can illuminate others, creating cascading paths.

3. Level Design

- **Grid-Based**: Levels are small, grid-based scenes with walls, mirrors, and the emitter/receiver.
- **Progression**: Early levels introduce basic reflection; later levels require multi-step chains or splitting beams (using special "splitter" mirrors).

4. Player Controls

- **Movement**: Arrow keys to navigate.
- **Interaction**: Press "A" to rotate a mirror 90 degrees clockwise.
- **Reset**: Press "B" to reset the level.

5. Win/Lose Conditions

- **Win**: All required mirrors are lit, and the beam reaches the receiver.
- **Lose**: Player runs out of moves (e.g., 5 rotations per level) or the beam fails to reach the receiver.

Implementation in GB Studio

Assets Needed

- **Sprites**:
  - Light emitter (static).
  - Light receiver (static/animated when activated).
  - Mirror (4-directional sprite sheet).
  - Light beam (animated line sprite).
  - Player (simple avatar).
- **Backgrounds**: Tiled walls and floors.
- **UI**: Move counter, reset button.

Event Scripts

1. **Emitter (On Init)**:
   - Launch a light beam actor in a fixed direction using `Launch Projectile` (set to "destroy on hit" with walls/mirrors).
   - Use a variable (e.g., `$BeamActive`) to track if the beam is active.
2. **Mirrors (On Hit: Light Beam)**:
   - If hit by the beam, change the mirror's sprite to the "lit" version using `Set Actor Animation State`.
   - Reflect the beam's direction based on the mirror's current rotation:
     - Use `Launch Projectile` to spawn a new beam actor in the reflected direction.
     - Play a sound effect (`Play Sound Effect`) on reflection.
3. **Mirrors (On Interact: Player)**:
   - Rotate the mirror 90 degrees using `Set Actor Direction`.
   - Decrement the move counter variable (`$MovesLeft`).
   - If `$MovesLeft <= 0`, trigger a level reset.
4. **Receiver (On Hit: Light Beam)**:
   - Change the receiver's sprite to "activated."
   - Check if all required mirrors are lit (using variables like `$Mirror1Lit`, `$Mirror2Lit`).
   - If all are lit, unlock the exit door (e.g., `Set Actor Sprite Sheet` to open door) and `Change Scene` to the next level.
5. **Player (On Update)**:
   - If `$MovesLeft <= 0` and the receiver is unlit, display a "Game Over" message and offer a reset.
6. **Triggers**:
   - Place a trigger over the exit door. If the receiver is lit, `Change Scene` to the next level when the player steps on it.

Variables

- `$BeamActive`: Boolean to toggle beam visibility.
- `$MovesLeft`: Counter for remaining rotations.
- `$Mirror1Lit`, `$Mirror2Lit`, etc.: Track which mirrors are activated.

Example Level Flow

1. Player enters the level. The emitter fires a beam to the right.
2. Player rotates Mirror A to reflect the beam upward.
3. The beam hits Mirror B, reflecting left toward the receiver.
4. Receiver activates, unlocking the exit.

Novelty and Depth

- **Puzzle Design**: Players must think ahead to create chains of reflections, especially in later levels with splitters or moving mirrors.
- **Visual Feedback**: Mirrors glow when lit, and the beam's path is clearly visible.
- **Replayability**: Limited moves force players to optimize their solutions.

Why It Works

- **Simple Controls**: Only movement and rotation.
- **Scalable Difficulty**: Easy to learn, hard to master.
- **Clever Twist**: Mirrors activating only when lit adds a layer of strategy beyond typical reflection puzzles.
- **GB Studio-Friendly**: Uses core features like actors, collision, variables, and event scripts without complex coding.